Modbus Wrapper Settings Guide

ACFA PSIM 1.0

Last update 08/22/2022

Table of Contents

1 List of terms used in Modbus Wrapper Settings Guide

Modbus is a communication protocol which is based on the master-slave architecture. To transfer the data, it uses the interfaces such as RS-485, RS-422, RS-232 (Modbus RTU protocol), and also Ethernet of TCP/IP network (Modbus TCP protocol).

Modbus Device is the automation system device (controller, sensor, operating mechanism) supporting the Modbus protocol.

Modbus Register is the Modbus protocol data type.

Modbus Rule is the *ACFA PSIM* microprogram used for processing the *BACnet* property settings.

2 Introduction into Modbus Wrapper Settings Guide

On the page:

2.1 Purpose of the document

The *Modbus Wrapper Module Settings Guide* is a reference and information guide meant for *Modbus Wrapper* configuration specialists.

This Guide presents the following materials:

1. general information about the *Modbus Wrapper* module;
2. *Modbus Wrapper* module settings;
3. working with the *Modbus Wrapper* module.

2.2 General information about the Modbus Wrapper integration module

The *Modbus Wrapper* integration module can carry out data exchange, get events and send commands to Modbus TCP or Modbus RTU protocols.

3 Supported hardware and licensing of the Modbus Wrapper integration module

The *Modbus Wrapper* module is licensed for 1 IP device.

Systems which operation is guaranteed by *Modbus Wrapper* universal integration are as follows:

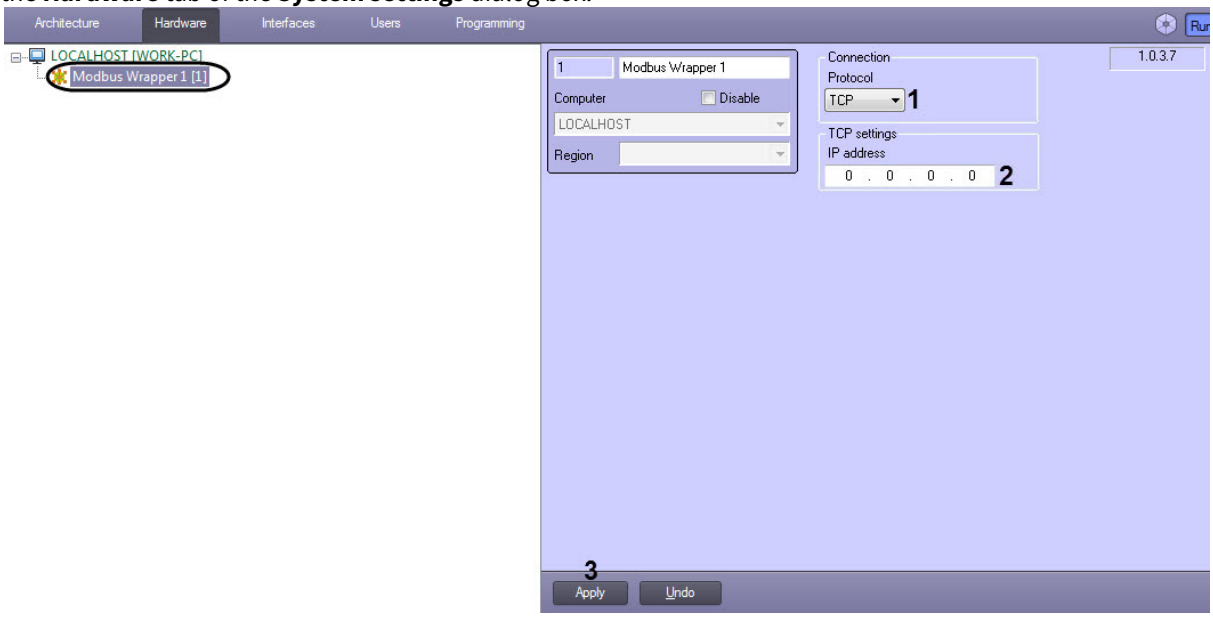
Name	**Integration capabilities in Axxon PSIM software**
Teletek IRIS Addressable Fire Alarm Panel (vendor: Teletek Electronics)	• Getting information on detection tool states, inputs/outputs, the panel itself, zones, and control commands
KSPA 9030-01 automatic fire fighting systems controller (vendor: PJSC «Gazprom avtomatizatsiya»)	• Getting information on detection tool states
KZ fire fighting controller (vendor: OOO «Vega-GAZ»)	• Getting information on detection tool states, inputs/outputs, and control commands
Schrack Seconet fire alarm stations (vendor: Schrack Seconet AG)	• Getting information on detection tool states
OptaSense perimeter intrusion detection system (vendor: OptaSense Ltd)	• Getting information on detection tool states

4 Configuration of the Modbus Wrapper integration module

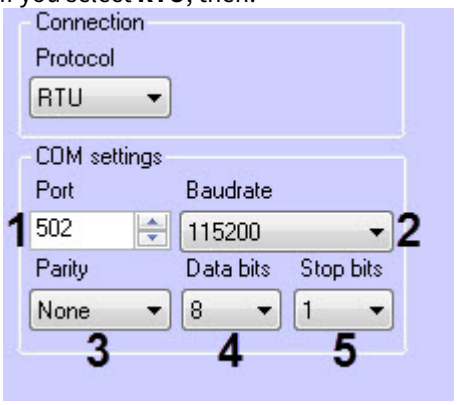
4.1 Configuring the Modbus Wrapper connection to ACFA PSIM

The *Modbus Wrapper* connection to *ACFA PSIM* is configured as follows:

1. Go to the **Modbus Wrapper** object settings panel which is created on the basis of the **Computer** object on the **Hardware** tab of the **System settings** dialog box.



2. From the **Protocol** drop-down list (**1**), select the protocol, above which the Modbus protocol will operate:
   - **TCP** - Modbus TCP.
     If you select **TCP**, enter the local IP-address of the *ACFA PSIM* Server in the **IP address** field (**2**).
   - **RTU** - Modbus RTU.
     If you select **RTU**, then:



   i. Enter the Modbus device connection COM-port in the **Port** field (**1**).
   ii. From the **Baudrate** drop-down list (**2**), select the data transfer speed over COM-port in bits per second.
   iii. From the **Parity** drop-down list (**3**), select the control type by parity:
      - **None** - control is disabled.
      - **Odd** - odd parity.
      - **Even** - even parity.
   iv. From the **Data bits** drop-down list (**4**), select the data format in bits: **7** or **8**.

v. From the **Stop bits** drop-down list (**5**), select the quantity of stop bits: **1, 1.5**, or **2**.

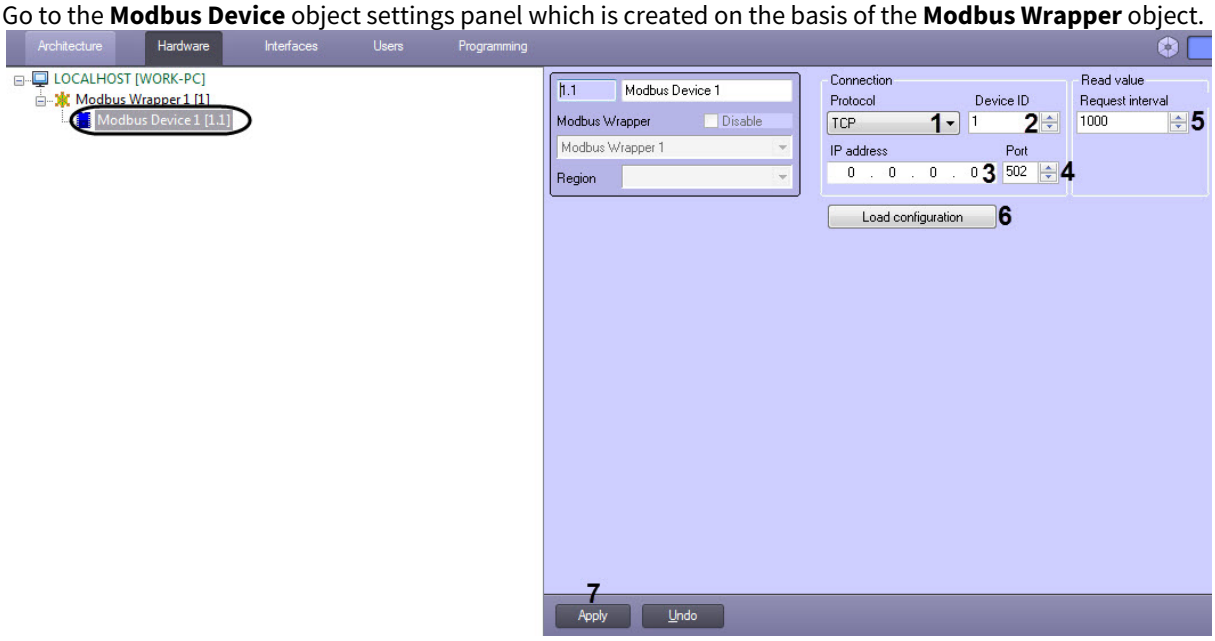
3. Click **Apply** (**3**) to save changes.

The *Modbus Wrapper* connection to *ACFA PSIM* is now configured.

4.2 Configuring the Modbus device

The *Modbus* device is configured as follows:

1. Go to the **Modbus Device** object settings panel which is created on the basis of the **Modbus Wrapper** object.



2. From the **Protocol** drop-down list (**1**), select the type of device connection to the *ACFA PSIM* Server.
   - **TCP** - connecting by Ethernet.
   - **RTU** - connecting through COM-port.
3. In the **Device ID** field (**2**), enter the device identification number which is specified in the device settings (Slave ID or Unit ID).
4. If you select **TCP** protocol, enter the IP address and device port in the **IP address** (**3**) and **Port** (**4**) fields respectively.
5. In the **Request interval** field (**5**), enter the polling period and new data reading from the device registers in milliseconds.
6. Click the **Load configuration** button (**6**) to build the tree of hardware and registers if there is a file of the Modbus register map in the **.mbmap** format. After clicking this button, it is necessary to select the corresponding file in the default window.
7. Click **Apply** (**7**) to save changes.

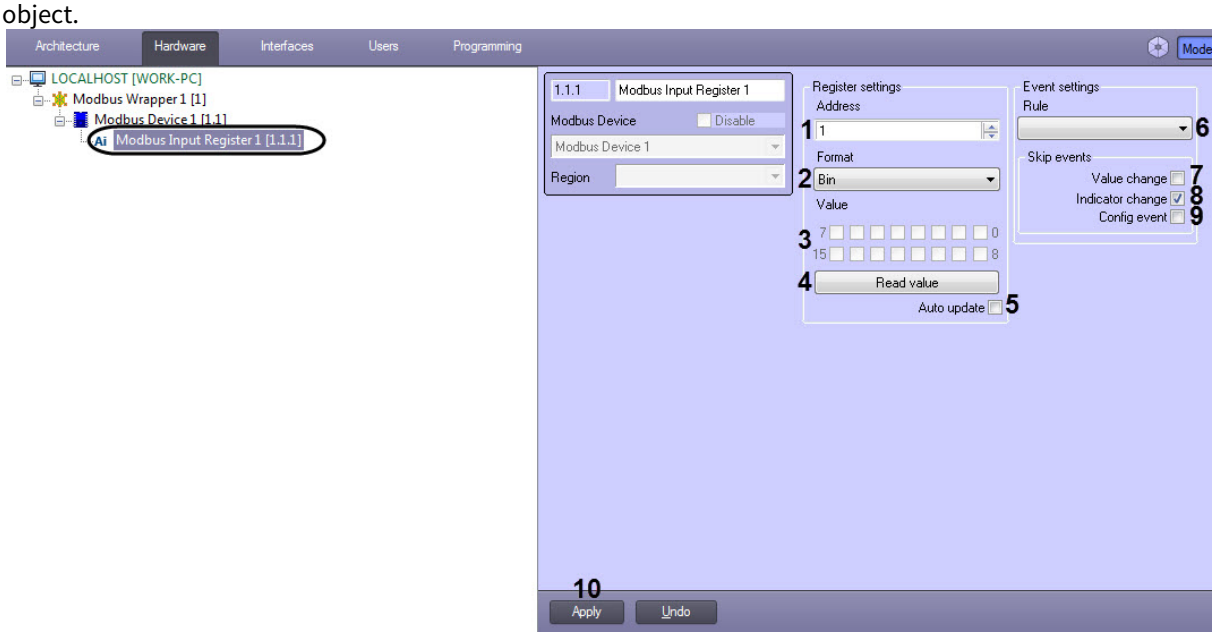
The *Modbus* device is now configured.

4.2.1 Setting up the Modbus Input Register

> **Note**
>
> By *Modbus* Input Register, the Analog Input is meant. It is possible to only read the Analog Input by getting the state of this input. The register size is 2 bytes.

The *Modbus* Input Register is configured as follows:

1. Go to the **Modbus Input Register** object settings panel which is created on the basis of the **Modbus Device** object.



2. In the **Address** field (**1**), enter the Input Register address in the register map of this device.

> **Note**
>
> Register map is provided by the vendor.

3. From the **Format** drop-down list (**2**), select the register data format:
   - **Bin** - binary.
   - **Dec** - decimal.
   - **Hex** - hexadecimal.

> **Note**
>
> The **Value** area (**3**) displays the current Input Register value in the selected format.

4. Click the **Read value** button (**4**) if it is necessary to read the device register.
5. Set the **Auto update** checkbox (**5**) if it is necessary to automatically read the register values with the interval specified on the **Modbus Device** object settings panel.
6. From the **Rule** drop-down list (**6**), select the rule to which this Input Register will obey (see Configuring the Modbus Rule).
7. Set the **Value change** checkbox (**7**) if it is not necessary to display the events on the register value change in the *Event Log*.
8. Uncheck the **Indicator change** checkbox (**8**) if it is necessary to allow displaying the register values in the plain text on the map.

> **Attention!**
>
> Value change of this parameter will be applied after restarting the *ACFA PSIM* Server.

9. Set the **Config event** checkbox (**9**) if it is not necessary to display the events in the *Event Protocol* when the rule is triggered.
10. Click **Apply** (**10**) to save the changes.

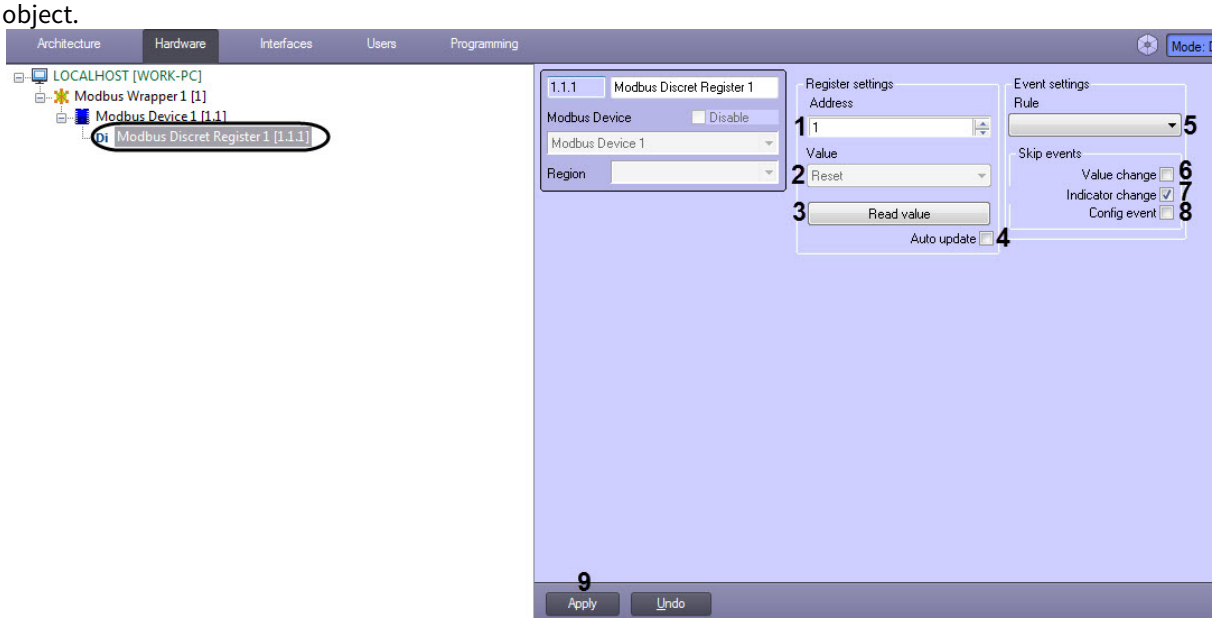
The *Modbus* Input Register is now configured.

4.2.2 Setting up the Modbus Discrete Register

> **Note**
>
> By *Modbus* Discrete Register, the Digital Input is meant. It is possible to only read the Digital Input by getting the real state of this input on sensor or device. The register size is 1 bit.

The *Modbus* Discrete Register is configured as follows:

1. Go to the **Modbus Discret Register** object settings panel which is created on the basis of the **Modbus Device** object.



2. In the **Address** field (**1**), enter the Discret Register address in the register map of this device.

> **Note**
>
> The register map is provided by the vendor.

3. Click the **Read value** button (**3**) if it is necessary to read the device register.

> **Note**
>
> The **Value** area (**2**) displays the current Discret Register value in the selected format.

4. Set the **Auto update** checkbox (**4**) if it is necessary to automatically read the register values with the interval specified on the **Modbus Device** object settings panel.
5. From the **Rule** drop-down list (**5**), select the rule to which this Register will obey (see Configuring the Modbus Rule).
6. Set the **Value change** checkbox (**6**) if it is not necessary to display the events on the register value change in the *Event Log*.
7. Uncheck the **Indicator change** checkbox (**7**) if it is necessary to allow displaying the register values in the plain text on the map.

> **Attention!**
>
> Value change of this parameter will be applied after restarting the *ACFA PSIM* Server.

8. Set the **Config event** checkbox (**8**) if it is not necessary to display the events in the *Event Protocol* when the rule is triggered.
9. Click **Apply** (**9**) to save the changes.

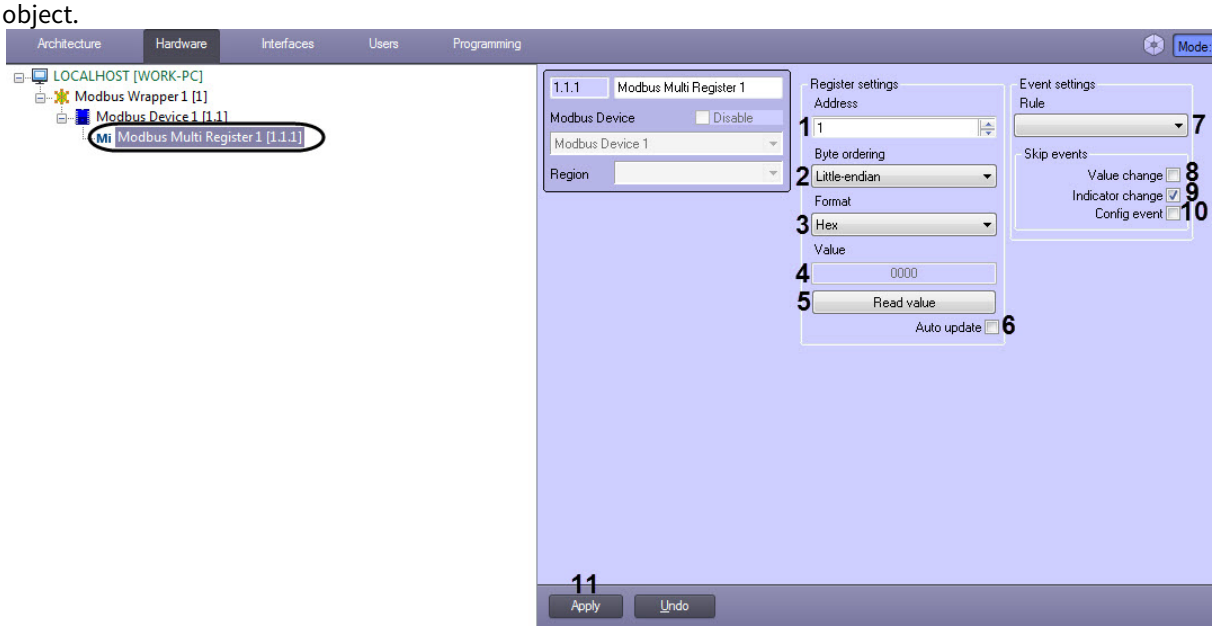
The *Modbus* Discret Register is now configured.

4.2.3 Setting up the Modbus Multi Register

> **Note**
>
> By *Modbus* Multi Register, the input consisting of two Modbus Input Registers is meant. It is possible to only read this input by getting its state. The register size is 4 bytes.

The *Modbus* Multi Register is configured as follows:

1. Go to the **Modbus Multi Register** object settings panel which is created on the basis of the **Modbus Device** object.



2. In the **Address** field (**1**), enter the Multi Register address in the register map of this device.

> **Note**
>
> The register map is provided by the vendor.

3. From the **Byte ordering** drop-down list (**2**), select the byte reading order of Multi Register: **Little-endian** or **Big-endian**.
4. From the **Format** drop-down list (**3**), select the register data format:
   - **Dec** - decimal.
   - **Hex** - hexadecimal.

> **Note**
>
> The **Value** area (**4**) displays the current Multi Register value in the selected format.

5. Click the **Read value** button (**5**) if it is necessary to read the device register.
6. Set the **Auto update** checkbox (**6**) if it is necessary to automatically read register values with the interval specified on the **Modbus Device** object settings panel.

7. From the **Rule** drop-down list (**7**), select the rule to which this Register will obey (see Configuring the Modbus Rule).
8. Set the **Value change** checkbox (**8**) if it is not necessary to display events on register value change in the *Event Log*.
9. Uncheck the **Indicator change** checkbox (**9**) if it is necessary to allow displaying register values in the plain text on the map.

   Attention!

   Value change of this parameter will be applied after restarting the *ACFA PSIM* Server.

10. Set the **Config event** checkbox (**10**) if it is not necessary to display events in the *Event Protocol* when the rule is triggered.
11. Click **Apply** (**11**) to save changes.

The *Modbus* Multi Register is now configured.

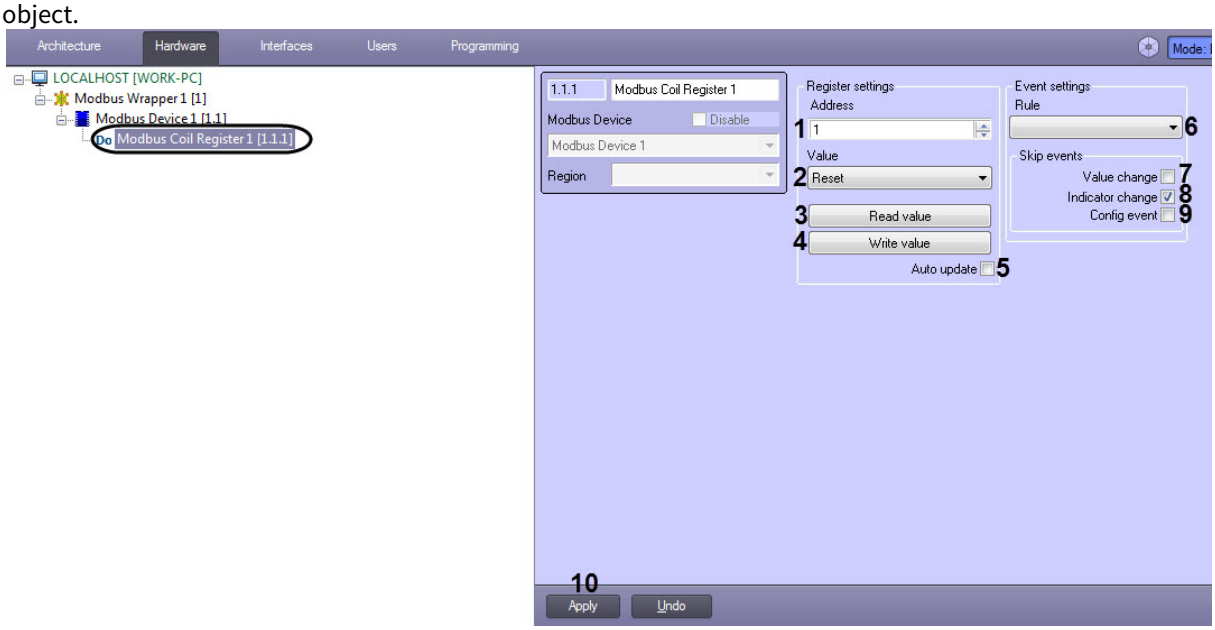
4.2.4 Setting up the Modbus Coil Register

Note

By *Modbus* Coil Register, the Digital Output, or Coil is meant. It is possible to either read or write the Digital Output. The register size is 1 bit.

The *Modbus* Coil Register is configured as follows:

1. Go to the **Modbus Coil Register** object settings panel which is created on the basis of the **Modbus Device** object.



2. In the **Address** field (**1**), enter the Coil Register address in the register map of this device.

   Note

   The register map is provided by the vendor.

3. From the **Value** drop-down list (**2**), select the coil register value: **Reset** or **Set**.
4. Click the **Read value** button (**3**) if it is necessary to read the device register.

5. Click the **Write value** button (**4**) if it is necessary to specify the selected value to the device register.
6. Set the **Auto update** checkbox (**5**) if it is necessary to automatically read the register values with the interval specified on the **Modbus Device** object settings panel.
7. From the **Rule** drop-down list (**6**), select the rule to which this Register will obey (see Configuring the Modbus Rule).
8. Set the **Value change** checkbox (**7**) if it is not necessary to display the events on the register value change in the *Event Log*.
9. Uncheck the **Indicator change** checkbox (**8**) if it is necessary to allow displaying of the register values in the plain text on the map.

> **Attention!**
>
> Value change of this parameter will be applied after restarting the *ACFA PSIM* Server.

10. Set the **Config event** checkbox (**9**) if it is not necessary to display the events in the *Event Protocol* when the rule is triggered.
11. Click **Apply** (**10**) to save the changes.

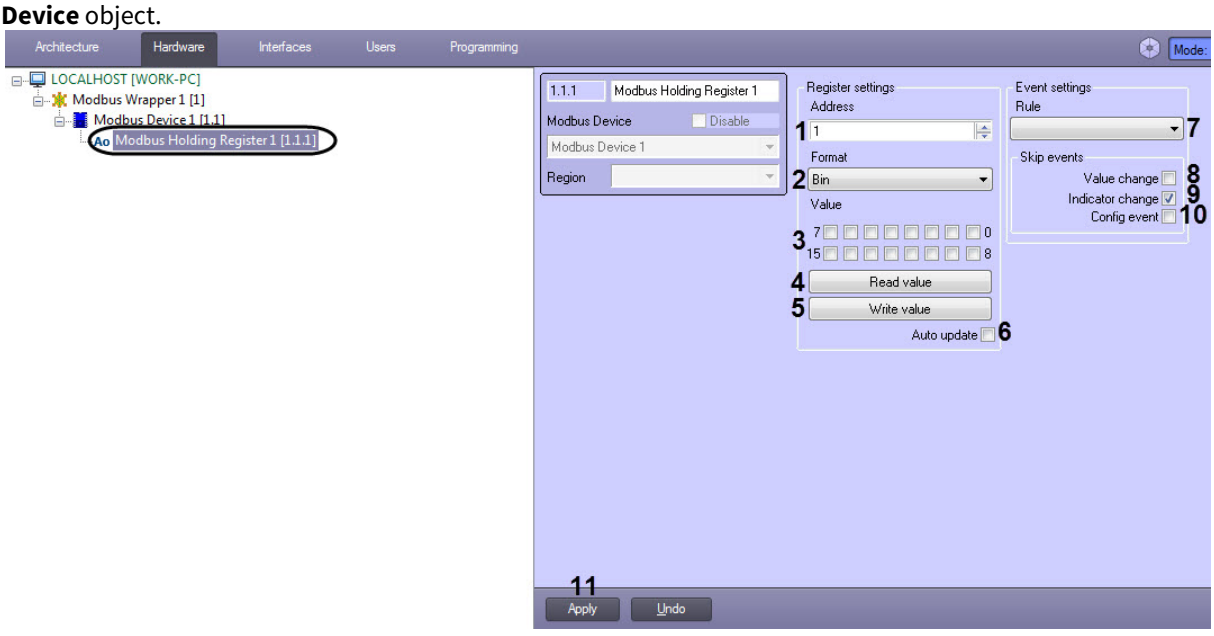
The *Modbus* Coil Register is now configured.

4.2.5 Setting up the Modbus Holding Register

> **Note**
>
> By *Modbus* Holding Register, the Analog Output or the Register containing some values which can be either written or read is meant. The register size is 2 bytes.

The *Modbus* Holding Register is configured as follows:

1. Go to the **Modbus Holding Register** object settings panel which is created on the basis of the **Modbus Device** object.



2. In the **Address** field (**1**), enter the Holding Register address in the register map of this device.

> **Note**

The register map is provided by the vendor.

3. From the **Format** drop-down list (**2**), select the register data format:
   - **Bin** - binary.
   - **Dec** - decimal.
   - **Hex** - hexadecimal.
4. The **Value** area (**3**) displays the current Register value in the selected format. You can also specify the value manually.
5. Click the **Read value** button (**4**) if it is necessary to read the device register.
6. Click the **Write value** button (**5**) if it is necessary to specify the selected value to the device register.
7. Set the **Auto update** checkbox (**6**) if it is necessary to automatically read the register values with the interval specified on the **Modbus Device** object settings panel.
8. From the **Rule** drop-down list (**7**), select the rule to which this Register will obey (see Configuring the Modbus Rule).
9. Set the **Value change** checkbox (**8**) if it is not necessary to display the events on the register value change in the *Event Log*.
10. Uncheck the **Indicator change** checkbox (**9**) if it is necessary to allow displaying of the register values in the plain text on the map.

    Attention!

    Value change of this parameter will be applied after restarting the *ACFA PSIM* Server.

11. Set the **Config event** checkbox (**10**) if it is not necessary to display the events in the *Event Protocol* when the rule is triggered.
12. Click **Apply** (**11**) to save changes.

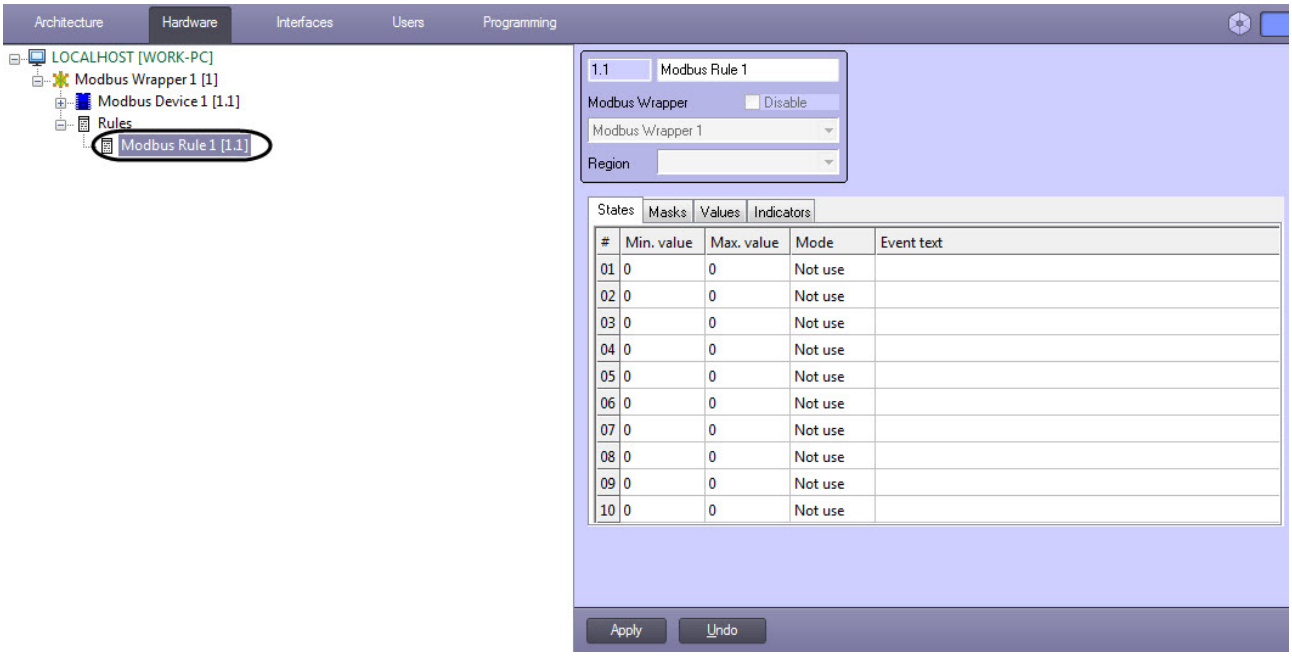
The *Modbus* Holding Register is now configured.

4.3 Configuring the Modbus Rule

Rules allow generating the events, changing the system states or indicator states in case of taking on the particular value by the Register.

Rules are configured on the **Modbus Rule** object settings panel which is created on the basis of the **Modbus Wrapper** object.

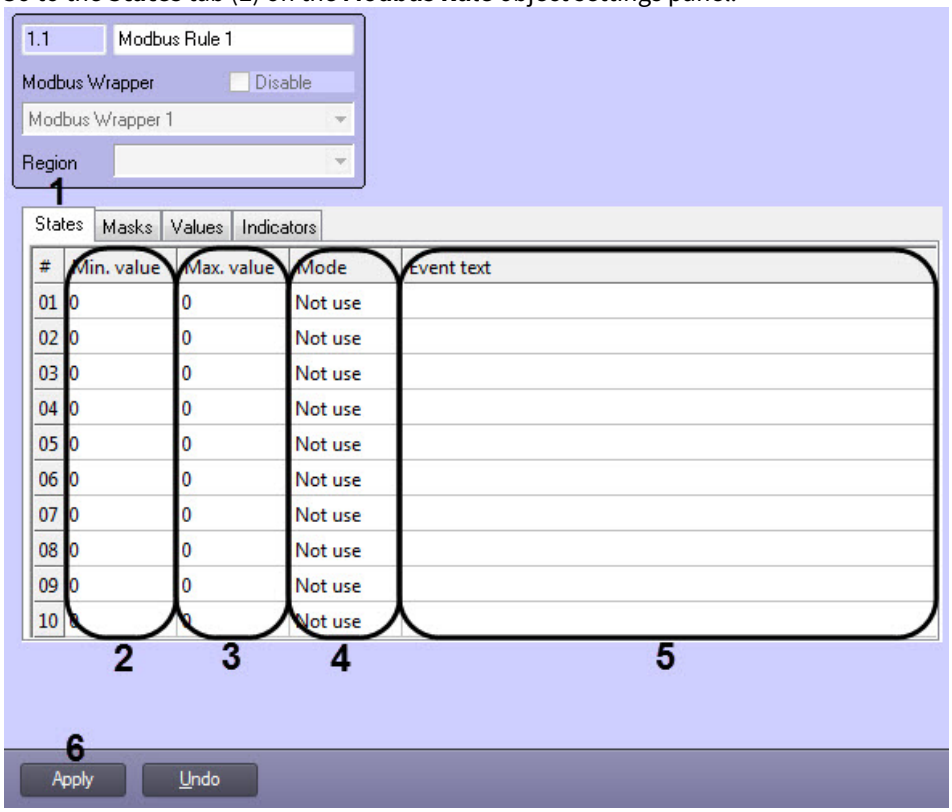
It is allowed to specify up to 10 conditions in every rule. Every condition corresponds to one state on the map (see Working with the Modbus Wrapper integration module).



4.3.1 Setting up the Modbus State Change Rule

The *Modbus* State Change Rule is configured as follows:

1. Go to the **States** tab (**1**) on the **Modbus Rule** object settings panel.



2. In the **Min. value** (**2**) and **Max. value** (**3**) columns, specify the register value range for which this condition will apply.

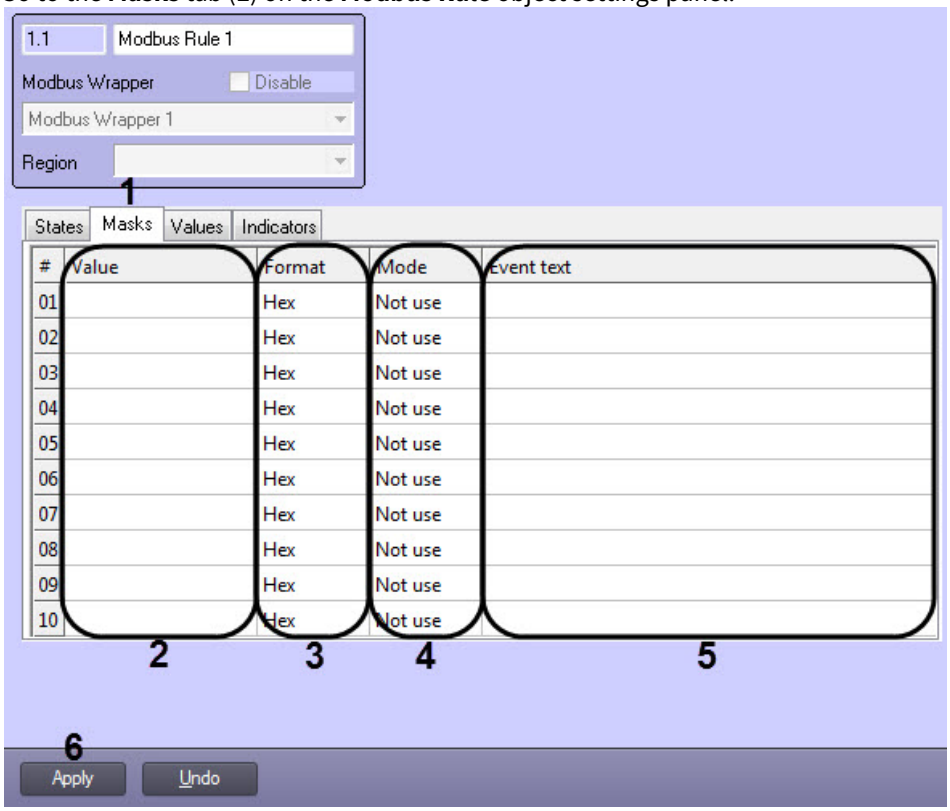
3. In the **Mode** column (**4**), select the action which will happen when the register takes on the value from the specified range:
    - **Off** - no action.
    - **State** - channel state change on the map.
    - **Event** - event generation.
    - **Both** - state change and event generation.
4. In the **Event text** column (**5**), enter the message which will be received when the register takes on the value from the specified range.
5. Click **Apply** (**6**) to save the changes.

The *Modbus* State Change Rule is now configured.

4.3.2 Setting up the Register State Change Rule using the Modbus Mask

The Register State Change Rule using the *Modbus* Mask is configured as follows:

1. Go to the **Masks** tab (**1**) on the **Modbus Rule** object settings panel.



2. In the **Value** column (**2**), enter the channel value.
3. In the **Format** column (**3**), enter the channel value format:
    - **Bin** - binary value format.
    - **Dec** - decimal value format.
    - **Hex** - hexadecimal value format.
4. In the **Mode** column (**4**), select the action which will happen when the register takes on the specified value:
    - **Off** - no action.
    - **State** - channel state change on the map.
    - **Event** - event generation.
    - **Both** - change state and event generation.
5. In the **Event text** column (**5**), enter the message which will be received when the register takes on the specified value.

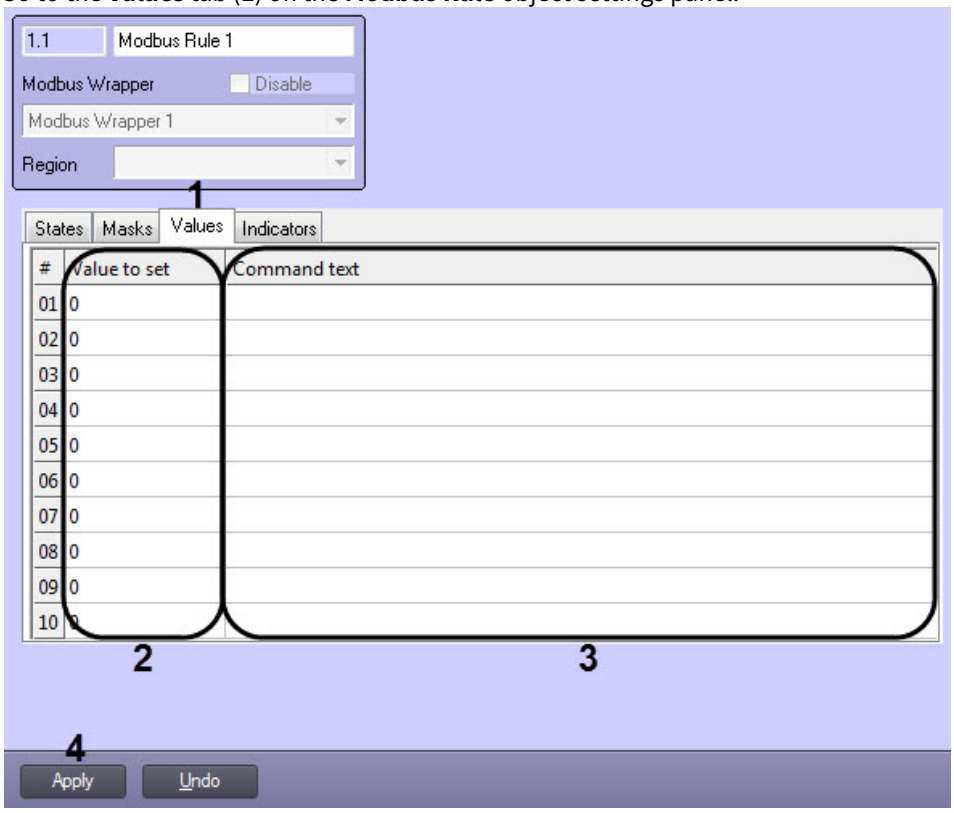
6. Click **Apply** (**6**) to save the changes.

The Register State Change Rule using the *Modbus* Mask is now configured.

4.3.3 Setting up the value assignment commands to Modbus Register

The value assignment commands to *Modbus* Register are configured as follows:

1. Go to the **Values** tab (**1**) on the **Modbus Rule** object settings panel.



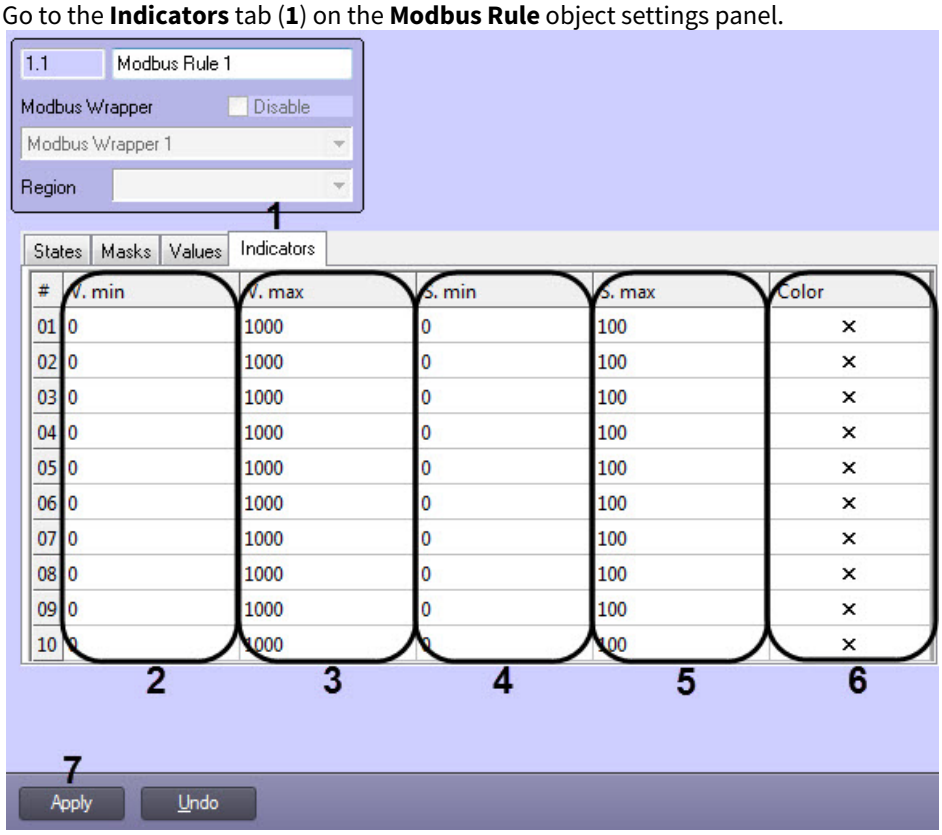
2. In the **Value** column (**2**), enter the channel value.
3. In the **Command text** column (**3**), enter the message which will be received when the register takes on the specified value.
4. Click **Apply** (**4**) to save the changes.

The value assignment commands to *Modbus* Register are now configured.

4.3.4 Setting up the Status Change Rule of Modbus Indicator

The Status Change Rule of *Modbus* Indicator is configured as follows:

1. Go to the **Indicators** tab (**1**) on the **Modbus Rule** object settings panel.



2. In the **V. min** (**2**) and **V. max** (**3**) columns, specify the register value range for which this condition will apply.

> **Attention!**
>
> If the register value falls into several ranges at once, then the indicator will take the value according to the condition with the smallest serial number of all the suitable conditions.

3. In the **S. min** (**4**) and **S. max** (**5**) columns, specify the indicator value range which it will take depending on the register value range.
4. In the **Color** column (**6**), select the indicator colors using the Windows default color palette.
5. Click **Apply** (**7**) to save the changes.

> **Note**
>
> The precise indicator value is calculated according to the following formula:
>
> $$S = \frac{(V - V.min)(S.max - S.min)}{V.max - V.min} + S.min$$
>
> , where V is the precise item value.

The Status Change Rule of *Modbus* Indicator is now configured.

5 Working with the Modbus Wrapper integration module

5.1 General information on working with the Modbus Wrapper module

The following interface objects are used for the *Modbus Wrapper* integration module operation:

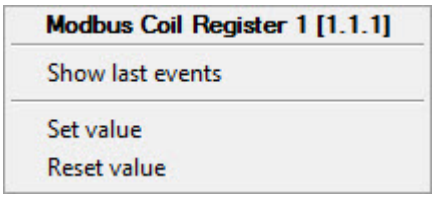
1. **Map**.
2. **Event Log**.

For detailed description of configuring these interface objects, please refer to the *Axxon PSIM* software package. Administrator's Guide.

For detailed description of using these interface objects, please refer to the *Axxon PSIM* software package. Operator's Guide.

5.2 Managing the Modbus Coil Register

The *Modbus* Coil Register is managed in the **Map** interactive window using the **Modbus Coil Register** object functional menu:



The *Modbus* Coil Register object functional menu commands description is given in the table.

Menu command	Function performed
Set value	Enables Digital Output
Reset value	Disables Digital Output

The *Modbus* Coil Register object can have the following states:

Modbus Coil Register 1 [1.1.1] N	Standard
Modbus Coil Register 1 [1.1.1] 1	State 1
Modbus Coil Register 1 [1.1.1] 2	State 2

Modbus Coil Register 1 [1.1.1] 	State 3
Modbus Coil Register 1 [1.1.1] 	State 4
Modbus Coil Register 1 [1.1.1] 	State 5
Modbus Coil Register 1 [1.1.1] 	State 6
Modbus Coil Register 1 [1.1.1] 	State 7
Modbus Coil Register 1 [1.1.1] 	State 8
Modbus Coil Register 1 [1.1.1] 	State 9
Modbus Coil Register 1 [1.1.1] 	State 10
Modbus Coil Register 1 [1.1.1] 	Not used

Note

The corresponding condition number of the rule is displayed on the channel state image.

The Register indicator takes on value and color according to the rule (see Configuring the Modbus Rule). If a new Register value does not meet any of the indicator rules, then it disappears.

If the channel value meets several states, the state image changes running through all the states. When you click on it, the smaller images of all the register states are displayed.

5.3 Managing the Modbus Holding Register

The *Modbus* Holding Register is managed in the **Map** interactive window using the **Modbus Holding Register** object functional menu:

The *Modbus* Holding Register object functional menu commands description is given in the table.

Menu command	Function performed
Change value	Sets the specified Analog Output value in the **Modbus Holding Register** object settings

The *Modbus* Holding Register states are similar to the *Modbus* Coil Register states (see Managing the Modbus Coil Register).

5.4 Managing the Modbus Device, Input Register, Discrete Register, and Multi Register

The *Modbus* Device, Input Register, Discrete Register, and Multi Register are not managed in the **Map** interactive window.

The *Modbus* Device can have the following states:

Modbus Device 1 [1.1] 	Disconnected
Modbus Device 1 [1.1] 	Connected

The *Modbus* Input Register, Discrete Register, and Multi Register states are similar to the *Modbus* Coil Register states (see Managing the Modbus Coil Register).